IN THE UNITED STATES DISTRICT COURT
FOR THE EASTERN DISTRICT OF TEXAS
MARSHALL DIVISION

VIRTAMOVE, CORP., Plaintiff, v. HEWLETT PACKARD ENTERPRISE COMPANY, Defendant.	Case No. 2:24-cv-00093-JRG (Lead Case) **JURY TRIAL DEMANDED**
VIRTAMOVE, CORP., Plaintiff, v. INTERNATIONAL BUSINESS MACHINES CORP., Defendant.	Case No. 2:24-CV-00064-JRG (Member Case) **JURY TRIAL DEMANDED**

PLAINTIFF VIRTAMOVE'S REPLY
CLAIM CONSTRUCTION BRIEF

TABLE OF CONTENTS

I. Claim terms for the '814 Patent.

A. "disparate computing environments" (claim 1)

VirtaMove's Proposed Construction	Defendants' Proposed Construction
environments run by standalone computers *alternatively:* environments <u>on</u> standalone computers <u>or on computers that are unrelated</u>	environments where computers are stand-alone or where there are multiple computers and where they are unrelated

Defendants do not dispute that their blind application of the specification's description of "disparate computing environments" would be confusing to the jury in the context of the claim language, because a jury may not understand what "environments ***where***…" means in context. As VirtaMove explained in its opening brief, "'where' is a reference to where the environment is…, not what the environment contains." Dkt. 143 at 3. Defendants do not dispute this in their briefing (and in fact agree that the specification discusses that an "environment" is "***on*** a computer," Dkt. 151 at 4) such that clarifying the scope of the claim to a jury is appropriate.

Defendants' responsive brief raises only two concerns with VirtaMove's briefing.

<u>First</u>, Defendants argue that a disparate environment is not "'***run by***' … computers," but is instead "an environment ***on*** a computer." Dkt. 151 at 4 (emphases in original). This is a distinction without a difference. Of course, something that is ***run by*** a computer (such as an app) is commonly referred to as ***on*** that computer, and Defendants give no plausible explanation why "run by" is less accurate than "on." Defendants confusingly assert that VirtaMove's construction "suggest[s] that the term is met by a single stand-alone computer simply running an environment" (Dkt. 151 at 4), but of course this would not meet the definition of environment**<u>s</u>** (plural) running on standalone computer**<u>s</u>** (plural). However, to the extent Defendants believe there is some difference in claim scope between environments **<u>run by</u>** a computer and environments **<u>on</u>** a computer, VirtaMove has no objection to making clear that the disparate computing environments

must be "on" the computers as Defendants contend the term would be understood.

Also regarding this language, Defendants inaccurately contend that "VirtaMove has not provided any justification for why the Court should deviate from the patent's own definition." Defendants do not even ***attempt*** to dispute that blindly importing the specification's description of "disparate computing environments" into the claims would lead to absurd results, as VirtaMove made clear in its opening brief. Defendants assert that "the 'environment' as used in the '814 patent… is clearly not directed to a planetary environment." Dkt. 151 at 6. But ***Defendants' construction*** does not make that clear in any way.

As a specific, less hyperbolic example of where there is a clear deviation between Defendants' proposed construction and Defendants' statement of how a POSITA would understand claim scope, Defendants contend that "the stand-alone, unrelated nature of multiple environments" is critical to understanding the claim scope. Dkt. 151 at 6.[1] But Defendants' proposed use of "where" does not clearly capture this stand-alone, unrelated nature. For example, both "Marshall, Texas" and "Judge Gilstrap's chambers" are two "environments where computers are stand-alone," because both of those environments include multiple stand-alone computers. However, those two environments are neither "stand-alone" nor "unrelated" to one another, because Judge Gilstrap's chambers are located ***within*** Marshall. Defendants cannot plausibly contend that, despite Marshall and Judge Gilstrap's chambers being related to one another, they would ***not*** be "disparate computing environments" under a blind, literal application of their claim construction. VirtaMove's construction does not deviate from the definition; it applies it in a clear

[1] For the record, VirtaMove agrees with Defendants in this understanding of what the concept of "disparate computing environments" is getting at; VirtaMove simply disagrees with Defendants that this concept is clearly reflected by Defendants' proposed construction because of the ambiguous use of the term "where," which can be easily clarified by making clear that "where" is a reference to the location of the environment, rather than its contents.

manner readily understandable to a jury.

Second, Defendants criticize VirtaMove's construction for omitting a possibility that computers may be "unrelated." Dkt. 151 at 5. VirtaMove is surprised by this disagreement given that other Defendants have stated that in the context of the claims, computers would not be "unrelated." *See* Dkt. 143 at 2 (citing Ex. 2 at 4, Ex. 3 at 6-7, and Ex. 4 at 2). However, VirtaMove does not believe that the addition of language regarding "unrelated" computers has a material impact on claim scope, and would not oppose a construction that references the fact that disparate computing environments could be satisfied by environments on (or run by) unrelated computers.

To alleviate Defendants' purported concerns, VirtaMove alternatively proposes a construction of "disparate computing environments" as "environments on standalone computers or on computers that are unrelated." This construction clarifies to a jury that "where" is a reference to where the environments are located, addresses both of Defendants' criticisms of VirtaMove's construction, and more clearly reflects even Defendants' understanding that a "stand-alone, unrelated nature of multiple environments" is required. *See* Dkt. 151 at 6.

B. "system files" (claims 1, 10)

Defendants do not dispute that "system files" are "filed provided with an operating system and which are available to applications as shared libraries and configuration files." Defendants ***also*** do not dispute that each individual file within these "system files" is a "system file." Thus, clearly, any library file provided with an operating system and available to an application is a "system file." And two such files would constitute "system files."

Defendants imply that only library files and configuration files ***together*** can constitute "system files." *See* Dkt. 151 at 7-8 ("the express definition of system files… requires ***both*** 'shared libraries ***and*** configuration files") (emphases in original). The specification, however, ***explicitly disproves*** Defendants' assertion: it states that "Linux Apache uses the following shared libraries,

supplied by the OS distribution, which are 'system' files." Ex. 1 at 2:55-3:5.

Defendants' proposed construction should be rejected, because Defendants intend to use that construction to inaccurately suggest to the jury that "system files… requires ***both*** 'shared libraries ***and*** configuration files.'" *See* Dkt. 151 at 7-8. As noted above, the patent makes clear that even a plurality of library files constitute "system files," contrary to Defendants' proposed claim interpretation. Ex. 1 at 2:55-3:5 ("the following shared libraries… are 'system' files").

To the extent the Court believes a construction would be helpful to the jury, VirtaMove believes that a more accurate construction of "system files," consistent with plain meaning and the specification, would be: "files provided with an operating system and which are available to applications as shared libraries and***/or*** configuration files." This alternative construction makes clear all that is required for "system files" is a plurality of items meeting the definition of a "system file," such that even a plurality of shared libraries would constitute "system files" in accordance with the teachings of the specification. *See* Ex. 1 at 2:55-3:5.

II. Claim terms for the '058 Patent.

A. "critical system elements" / "operating system critical system elements" / "Shared library critical system elements" (claim 1)

As VirtaMove explained in its opening brief, Dr. Stavrou failed to analyze the question of whether a service is "critical" to "a software application" in the context of the recited "operating system" and the recited "application." Dkt. 143 at 7. Indeed, Dr. Stavrou was ***explicit*** that he arrived at his conclusion that "critical" is indefinite on the basis that the '058 Patent does not identify the "software applications" or "operating systems" relevant to the analysis, such that a POSITA would not know what applications and operating systems to "consider":

> 38. Thus, the "critical" quality of any given service would not only depend on the software applications a POSITA may consider, but also on the operating systems, the universe of which is not defined by the '058 Patent. Again, the specification and prosecution history of the '058 Patent provide no objective standard to guide a POSITA's determination of what makes a service "critical."

Ex. 6 to Dkt. 143 ("Stavrou Decl.") at ¶38.

Dr. Stavrou's testimony is plainly insufficient to show indefiniteness, because in the context of the proposed construction, a POSITA ***would*** know the relevant "software application" being evaluated as well as the relevant "operating system[]"—they are the recited "software application" and "operating system." *See* Dkt. 143 at 7. Defendants present no serious defense to this critical flaw in Dr. Stavrou's analysis and conclusion regarding indefiniteness.

Instead, Defendants incorrectly assert that Dr. Stavrou asserted that "even ***if*** a POSITA knew the specific software application or operating system in question, s/he still would have no objective way to determine whether a given software's use of that service is 'critical' as recited by the claims." Dkt. 151 at 13. This is ***not*** a quotation from Dr. Stavrou, and a review of Dr. Stavrou's declaration and deposition testimony makes clear that Dr. Stavrou did not specify what his conclusion on indefiniteness would be under a scenario where a POSITA ***did*** know the "specific software application [and] operating system in question." Instead, Dr. Stavrou testified that he believed (incorrectly) that a POSITA would ***not*** know these details in performing the evaluation because "the universe of [these details] is not defined by the '058 Patent." Stavrou Decl. ¶38.

Accordingly, Dr. Stavrou's testimony, based on the false premise that a POSITA would not know basic contextual information, is not evidence of indefiniteness.

Defendants' additional arguments are unavailing. For example, Defendants point to an IEEE Dictionary definition of "criticality" as it relates to a "description of the intended use an application of the system." Dkt. 151 at 11. But "critical" in the context of the claim is not about an "intended use of an application of the system," it is about a specific inquiry of whether a specific service is "critical" to a specific thing—the operation of a specific software application. None of Defendants' evidence even remotely suggests that a POSITA would not be able to determine whether a service is "critical" in this context.

Defendants next raise several potential ways to evaluate "critical," but present no evidence or allegation that any of these several ways would fall outside the scope of criticality. Dkt. 151 at 11-12. Defendants fail to consider that if an application relies on a service to perform an operation (an understanding that includes each of Defendants' hypotheticals), that service is "critical to" that operation. Instead, Defendants rely solely on inapposite dictionary definitions and the testimony of an expert who incorrectly assumed that the relevant application and operating system would be unknown. Defendants fall far short of their clear and convincing burden.

B. "functional replicas" (claim 1)

Defendants do not deny that the '058 patent defines "replica" to include "replicas or substantial functional equivalents or replacements of kernel functions." '058 Patent at 8:27-29 (cited, Dkt. 151 at 16-17). Rather than being "a hodgepodge of words" as Defendants contend, this language understandably conveys the definite scope of "functional replica."

First, although Defendants and their expert argue that the plain and ordinary meaning of "replica" refers to exact copies, they must concede that the patent does ***not*** use "replica" in that narrow sense. The statement that "The term replica, shall encompass any of these meanings," *i.e.* "replicas or substantial functional equivalents or replacements," is consistent with this, and can

readily be understood to mean: the term replica *as used in this patent* encompasses replicas *in the plain and ordinary sense* of an exact copy, as well as substantial functional equivalents and replacements. This is a definition, not a contradiction.

Defendants' argument that "functional replicas" must be logically narrower than the plain and ordinary meaning of "replica" is also incorrect. "Functional" defines and ***limits*** the metric that the "functional replica" must replicate. For example, if one component has the functionality of displaying text to a monitor, its "functional replica" must have a substantially equivalent function (in accord with the lexicography above). It need not be a "replica" in any sense ***other*** than functionality (e.g., it need not have a substantially equivalent name or implementation). "Functional replica" thus has a ***broader*** scope than "replica" without qualification.

The converse is that, at least in the digital realm where a "copy" is genuinely the same thing as the original, a "copy of a CSE" is *ipso facto* a substantial functional equivalent of that CSE. It cannot provide less or different functionality than itself. Thus "copy" is logically ***narrower*** than "substantial functional equivalent." It is for this reason that VirtaMove's proposed construction omits "replica," because including it would not change the claim scope.

Finally, Defendants can show no case holding that the determination of whether two things are "substantial functional equivalents" is subjective or lacks objective metrics. As VirtaMove has pointed out, juries regularly determine whether things are "equivalent," including whether they perform "substantially the same function." These are literally part of the jury instructions. Defendants' demand for additional "objective guidance" thus contradicts established law.

C. "forms a part of the one or more of the plurality of software applications" (claim 1)

As VirtaMove previously explained, Defendants' construction is a "requirement" to satisfy this limitation (Dkt. 143 at 10), but it is not a ***definition*** because not ***everything*** that "resides in the

same address space as" the application's application code "form[s] a part of the application." The file history does not state or imply to the contrary, and Defendants give no explanation for their attempt to broaden the scope of "forms a part..." beyond its plain and ordinary meaning.

Defendants' attempt to use alleged ***disclaimer*** as cover to improperly ***broaden*** claim scope is relevant because Defendants have asserted that prior art discloses this limitation merely through disclosures of a location in memory, such as "in random access memory," "on the file system," or in "a relational database structure." Ex. 1 at 14-15. A jury might interpret Defendants' construction to mean that anything in the same "memory" or "file system" as the application (but that is not a "proxy") thereby "literally forms a part" of the application, distorting the claim scope.

Defendants allude to a potential dispute about "whether a SLCSE can 'form a part of' an application if it runs in a context 'exclusive of the application.'" Dkt. 151 at 23. VirtaMove is not aware of any such dispute, but Defendant's construction only excludes "a *proxy* that is exclusive of the application" such that it would not resolve any such dispute. In short, both parties agree that the file history discussion impose a requirement on this limitation, but there is no evidence that it ***defines*** the limitation, such that Defendant's construction should be rejected.

D. "shared library" (claim 1)

Defendants argue that the patentee actually intended to define "shared library" to mean "a specific type of physical memory space" rather than a library stored within that memory space. *See* Dkt. 151 at 21. This misreading of the specification is inconsistent even with Defendants' own understanding of "shared library" and should be rejected. *See id.* (admitting that "code spaces… ***store*** shared libraries" and are not themselves shared libraries).

At the outset, Defendants' attempt to simply ignore the clear and unambiguous definition in the provisional application fails. The patent expressly incorporates the provisional by reference,

specifically for its teachings of shared libraries. '058 Patent at 8:7-11 ("Software applications 32a, 32b utilize ***shared libraries*** 34 as is done in U.S. Provisional Patent Application Ser. No. 60/512,103... which is incorporated by reference herein."). It thus forms part of the specification, at least for purposes of claim construction. *See Trustees of Columbia Univ. in City of New York v. Symantec Corp.*, 811 F.3d 1359, 1365–66 (Fed. Cir. 2016) (definition of "byte sequence feature" in provisional application supported construction of that term).

The longer definition in the specification does include two additional sentences not stated in the provisional definition. Defendants criticize VirtaMove for "truncating" those sentences away, but without giving any explanation how they modify claim scope. The second sentence ("The code space is different than that occupied by the kernel and its associated files.") is redundant of the requirement, included in VirtaMove's proposal, that the code space "is shared among all *user* mode applications," not the kernel. This is also consistent with the express claim language "for use by the plurality of applications in user mode." VirtaMove does not believe that including or excluding this sentence substantively affects claim scope.

The third sentence unambiguously ***confirms*** VirtaMove's interpretation, stating: "***The shared library files are placed in an address space*** that is accessible to multiple applications." This makes clear that a shared library is something ***placed in*** a "space," rather itself ***being*** a "space." Beyond that, the third sentence appears to impose a weaker version of the requirement, already reflected in the first sentence and in VirtaMove's proposal, that the shared library is located in a code space "shared among *all* user mode applications," not merely "multiple applications."

Defendants incorrectly suggest that the presence of SLCSEs within the shared library confirms that a library is a "physical memory space" by analogy to a "physical brick-and-mortar library." Dkt. 151 at 20. But this ignores the plain and ordinary meaning of "application library"

in this context, which is "A collection of functions in an archive format that is combined with an application to export system elements." Indeed, Defendants demand that "application library" be construed to have that very definition.[2] An SLCSE can plainly be stored within a collection in archive format. In the brick-and-mortar analogy, the "shared library" should be understood as the library itself, which is ***located*** in a physical space such as a plot of land (physical memory).

Even Defendants' own briefing betrays the correct understanding that a shared library is not ***itself*** a code space but is instead stored ***within*** a code space. *See* Dkt. 151 at 21 (admitting that "code spaces… ***store*** shared libraries" such that the library is ***within*** the code space). Only VirtaMove's construction accurately reflects this understanding (as well as the provisional application's express definition), and Defendants' proposal to the contrary should be rejected.

Dated: March 14, 2025

Respectfully submitted,

/s/ Reza Mirzaie
Reza Mirzaie
CA State Bar No. 246953
Marc A. Fenster
CA State Bar No. 181067
Neil A. Rubin
CA State Bar No. 250761
Jacob R. Buczko
CA State Bar No. 269408
James S. Tsuei
CA State Bar No. 285530
James A. Milkey
CA State Bar No. 281283
Christian W. Conkle
CA State Bar No. 306374
Jonathan Ma
CA State Bar No. 312773
Daniel Kolko
CA State Bar No. 341680

[2] VirtaMove does not believe "application library" needs to be construed, but also does not believe that the proposed construction affects claim scope.

RUSS AUGUST & KABAT
12424 Wilshire Boulevard, 12th Floor
Los Angeles, CA 90025
Telephone: 310-826-7474
Email: rmirzaie@raklaw.com
Email: mfenster@raklaw.com
Email: nrubin@raklaw.com
Email: jbuczko@raklaw.com
Email: jtsuei@raklaw.com
Email: jmilkey@raklaw.com
Email: cconkle@raklaw.com
Email: jma@raklaw.com
Email: dkolko@raklaw.com

Qi (Peter) Tong
TX State Bar No. 24119042
8080 N. Central Expy, Suite 1503
Dallas, TX 75206
Email: ptong@raklaw.com

ATTORNEYS FOR PLAINTIFF VIRTAMOVE, CORP.

CERTIFICATE OF SERVICE

I certify that this document is being served upon counsel of record for Defendants on March 14, 2025 via CM/ECF.

/s/ *Reza Mirzaie*